

Компьютерная графика Современные технологии компьютерной графики и рендеринга

Лекция 5

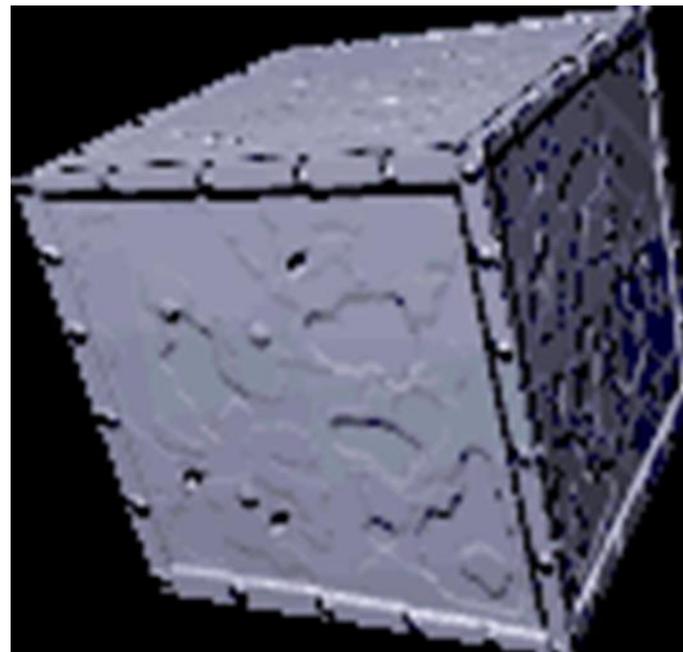
02.04.02 ФИИТ

Разработка мобильных приложений и компьютерных игр

2025-2026

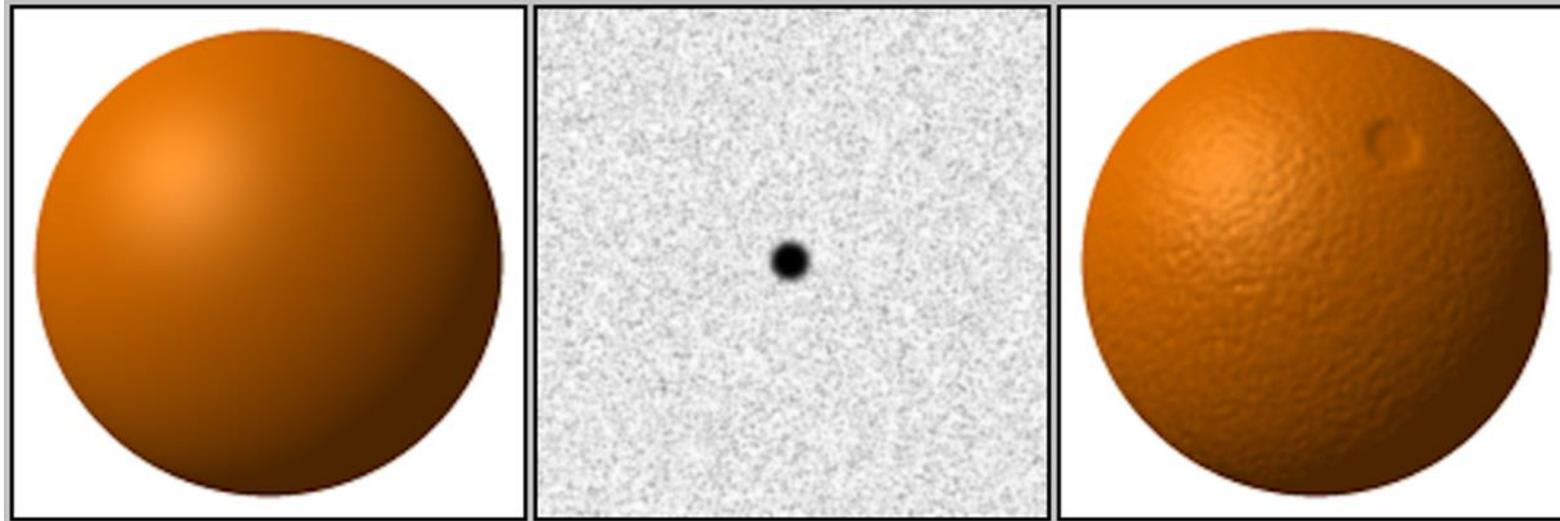
Рельефное текстурирование — что это?

Видим старинный металлический сундук(комод) или ящик
А моделью является просто кубик с шестью гранями



Рельефное текстурирование — это общее название для методов, которые добавляют иллюзию сложной геометрии на плоских или простых поверхностях.

В чем отличие от обычного текстурирования?



Рельефное текстурирование отражает реальное положение источника света в сцене и даже изменение его местоположения

Виды рельефного текстурирования

Bump mapping

Normal mapping

Displacement mapping

Parallax mapping

Parallax occlusion mapping

Немного из истории Bump mapping



Бампмаппинг был разработан Блинном (Blinn) еще в 1978 году

Jim Blinn



Джеймс Ф. Блинн 1949 (77) — американский учёный и пионер в области компьютерной графики, внёсший значительный вклад в развитие технологий визуализации и компьютерного моделирования. Известен как автор алгоритмов компьютерной графики, разработчик модели освещения Блинна–Фонга.

Альма-матер:

Университет Юты

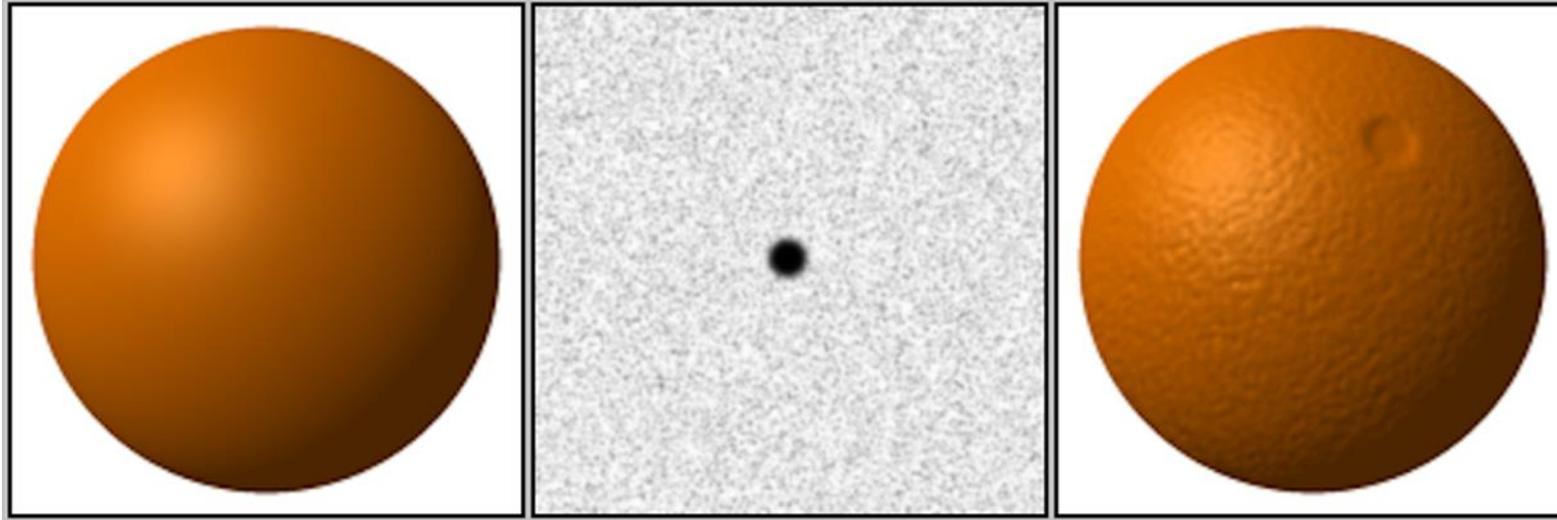
Мичиганский университет

Награды:

Стипендия Макартура (1991)

Медаль НАСА за выдающиеся заслуги

Что такое рельефная карта?



Рельефная карта (текстура) — это обычная **текстура**, только **в отличие** от первой, несущей информацию о **цвете** определенных участков, **рельефная карта** несет информацию о **неровностях**

Самый простой вид — карта высот, где яркость пикселя говорит о том, насколько эта точка выступает или углублена. Белый — высоко, черный — низко.

Применение Bump mapping

- Можно имитировать неровности и шероховатости
 - дерево
 - камень
 - шелушащуюся краску
 - и т.д.
- Нельзя имитировать крупные впадины и возвышенности

Что нужно?

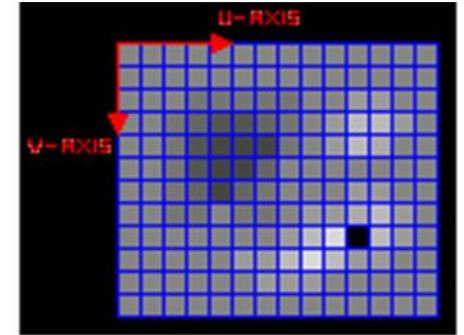
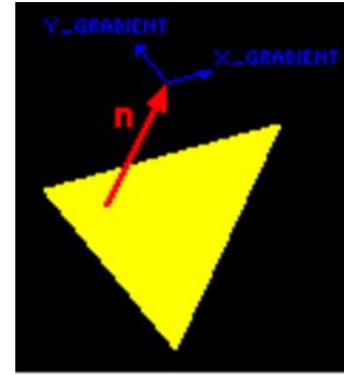
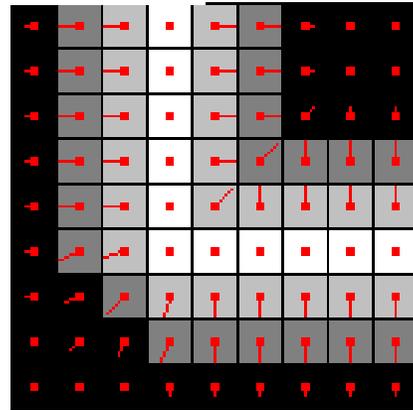
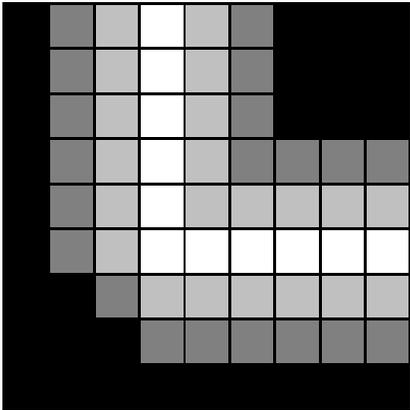
Преобразовать

информацию о высоте
неровностей на карте высот

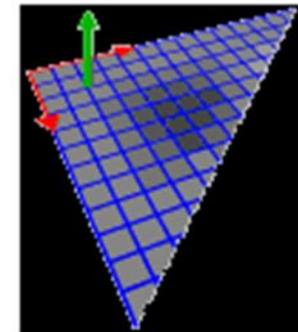
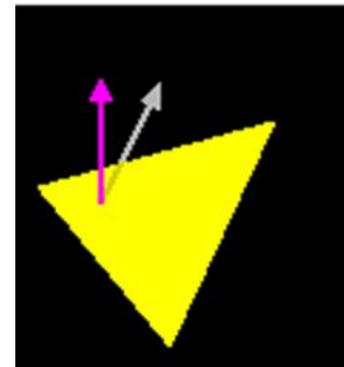


информацию о величине
подстройки вектора нормали

Как реализовать?



$$x_gradient = pixel(x-1, y) - pixel(x+1, y)$$
$$y_gradient = pixel(x, y-1) - pixel(x, y+1)$$



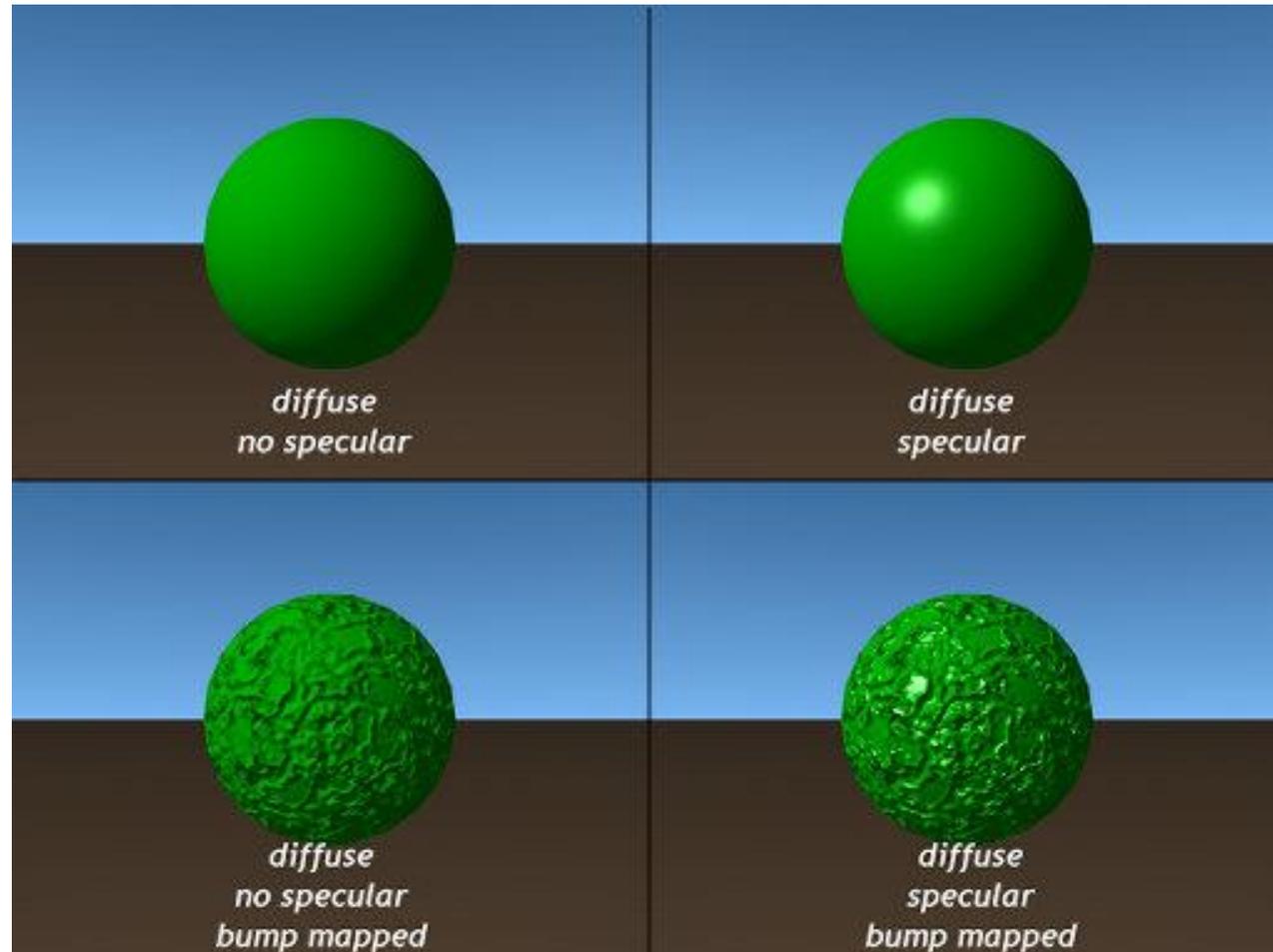
$$New_Normal = Normal + (U * x_gradient) + (V * y_gradient)$$

Bump mapping

Поверхность кажется неровной, хотя геометрически она плоская.

Этот метод был прорывом для своего времени, но у него есть недостатки.

Он не учитывает, что неровности могут отбрасывать тени друг на друга, и с близкого расстояния иллюзия может разрушаться



Bump mapping в играх

рендеринг без бампмаппинга

рендеринг с бампмаппингом без бликовой составляющей

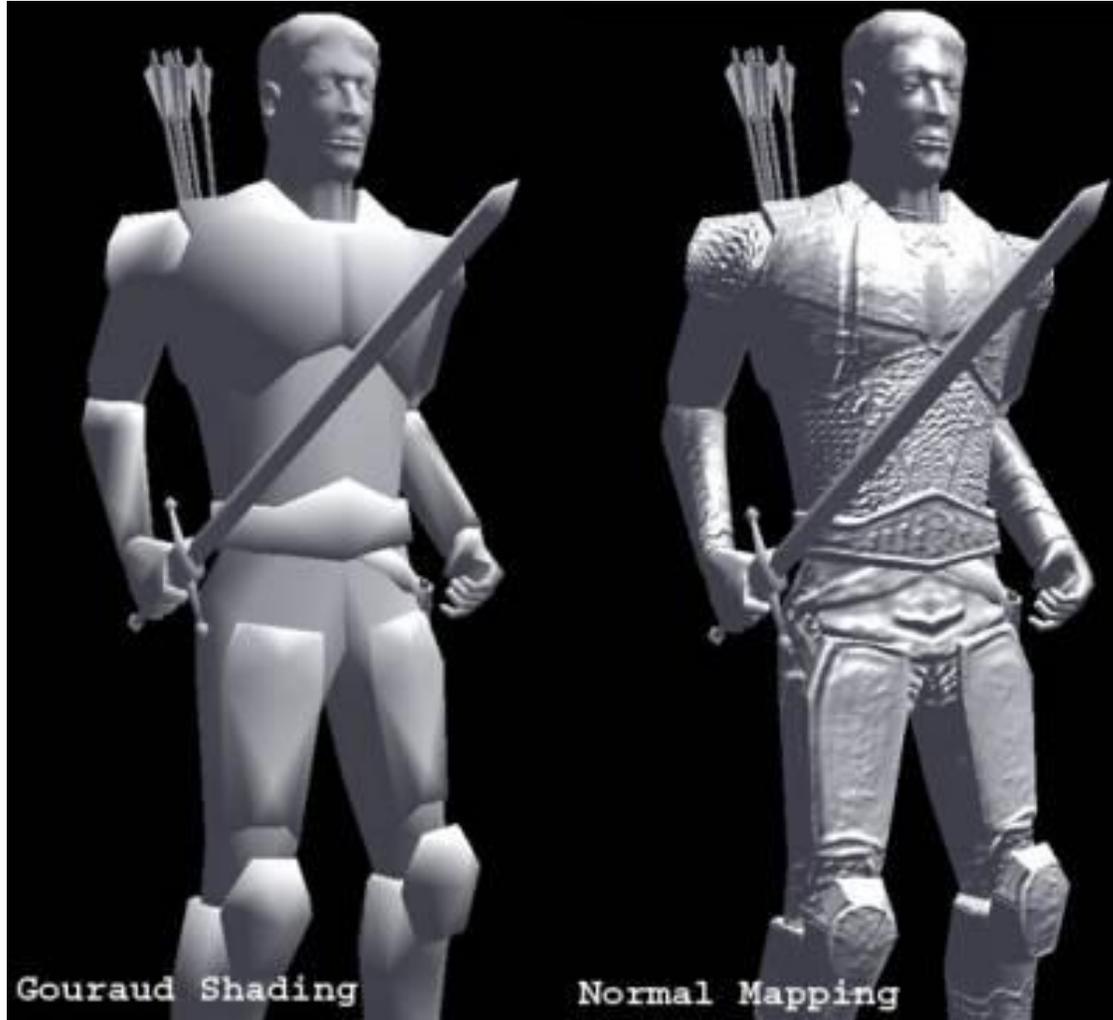


В играх bump mapping начали активно использовать в конце 90-х

Карта нормалей (normal map)

- Это дальнейшее развитие технологии бамп (bump)
- Иногда карты нормалей называют «3-х точечная бамп-карта» (Dot3 bump mapping)

Normal Mapping



Слева — низкополигональная модель с обычной текстурой.
Справа — та же модель, но с картой нормалей.

Детализация лица, морщины, поры кожи — все это создается не геометрией, а картой нормалей.

Глаз видит сложную поверхность, но GPU обрабатывает всего несколько тысяч треугольников.

История normal mapping — SIGGRAPH 1996

- Впервые идея о том, что можно «брать» детали с высокополигональной модели и переносить на низкополигональную, была предложена на SIGGRAPH в 1996 во время доклада «Использование сглаженных поверхностей для добавления плотности полигональной сетке» авторов Кришнамурати и Левой
- SIGGRAPH (The Special Interest Group for Computer Graphics)
- Тогда речь шла об использовании для **Displacement Mapping**
- Предлагалось использовать созданные на основе NURBS **карты смещения** (displacement maps) для увеличения детализации низкополигональной поверхности
- **Реализация появилась гораздо позже**

SIGGRAPH 1998 — Appearance Preserving Simplification

- В 1998 году опять на SIGGRAPH были представлены 2 доклада о перенесении деталей в виде **карт нормалей** от высокополигональных моделей в низкополигональные
- Один из них: Appearance Preserving Simplification (Упрощение с сохранением внешнего вида).
Авторы: Jonathan Cohen, Marc Olano, Dinesh Manocha

Они показали, как можно взять высокополигональную модель, упростить ее геометрию, но сохранить внешний вид, выполнив запекание деталей в карту нормалей.
Это стало **стандартом индустрии на десятилетия**.

- www.graphicon.ru/oldgr/library/siggraph/98/papers/cohen/cohen.pdf

Из статьи Appearance Preserving Simplification



Figure 1: Bumpy Torus Model. *Left*: 44,252 triangles full resolution mesh. *Middle and Right*: 5,531 triangles, 0.25 mm maximum image deviation. *Middle*: per-vertex normals. *Right*: normal maps

Appearance Preserving Simplification



249,924 triangles 62,480 triangles 7,809 triangles 975 triangles
0.05 mm max image deviation 0.25 mm max image deviation 1.3 mm max image deviation 6.6 mm max image deviation

Figure 12: Close-up of several levels of detail of the armadillo model. *Top*: normal maps *Bottom*: per-vertex normals

Общее: bump и normal -mapping

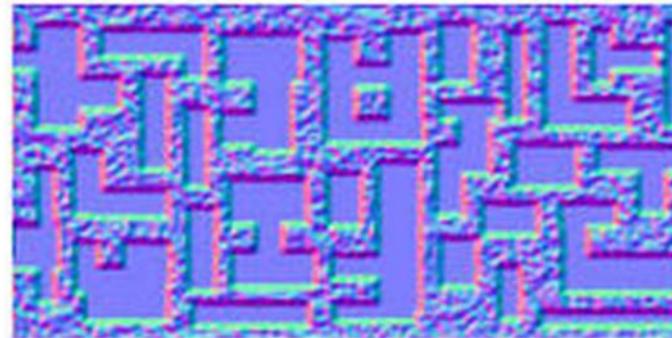
- Увеличивает визуальную детализацию 3d-модели без добавления дополнительных полигонов
- Карта рассчитывается для каждого пикселя текстуры

Различия: bump и normal -mapping

- Бамп рассчитывается на основе текстуры с одним каналом (grayscale — оттенки серого)
- Карты нормалей используют многоканальную RGB-текстуру, поэтому метод дает большую точность, чем Bump mapping



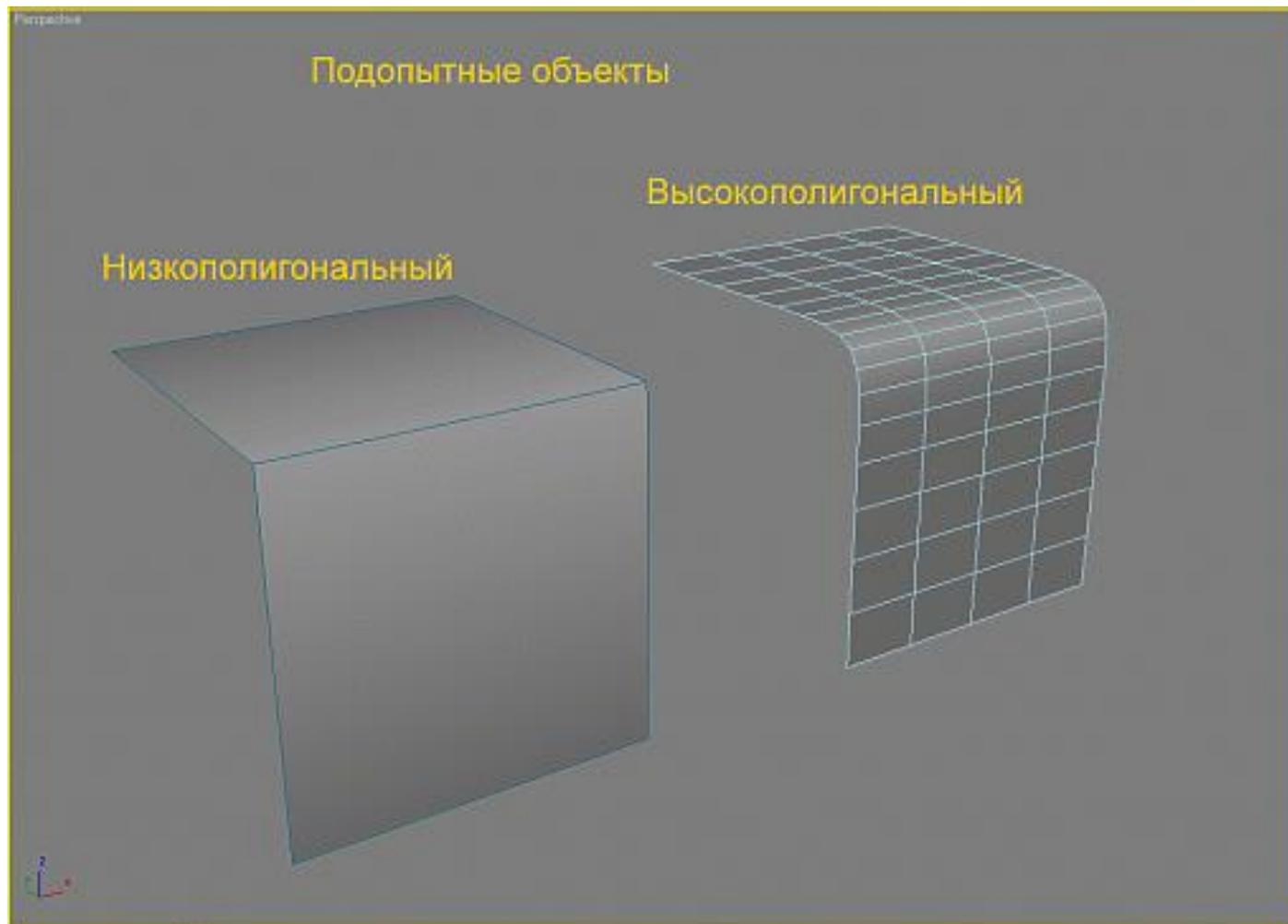
Bump Map



Normal Map

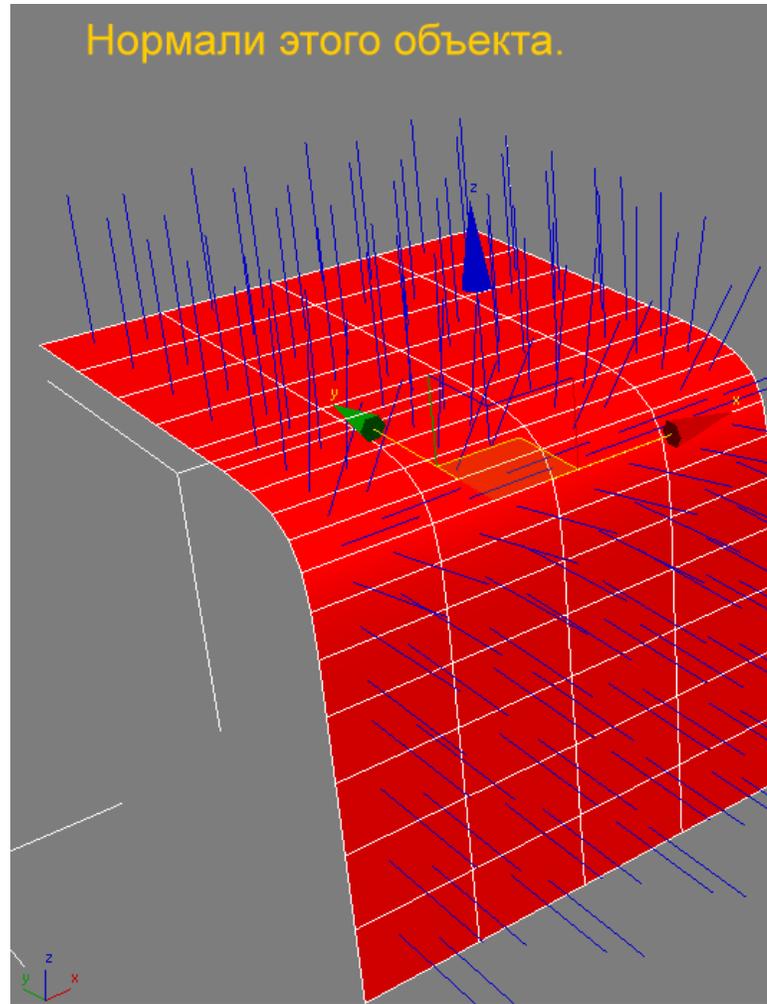
Нормал-маппинг. Начало

Низкополигональный кубик превращаем в редакторе в высокополигональный

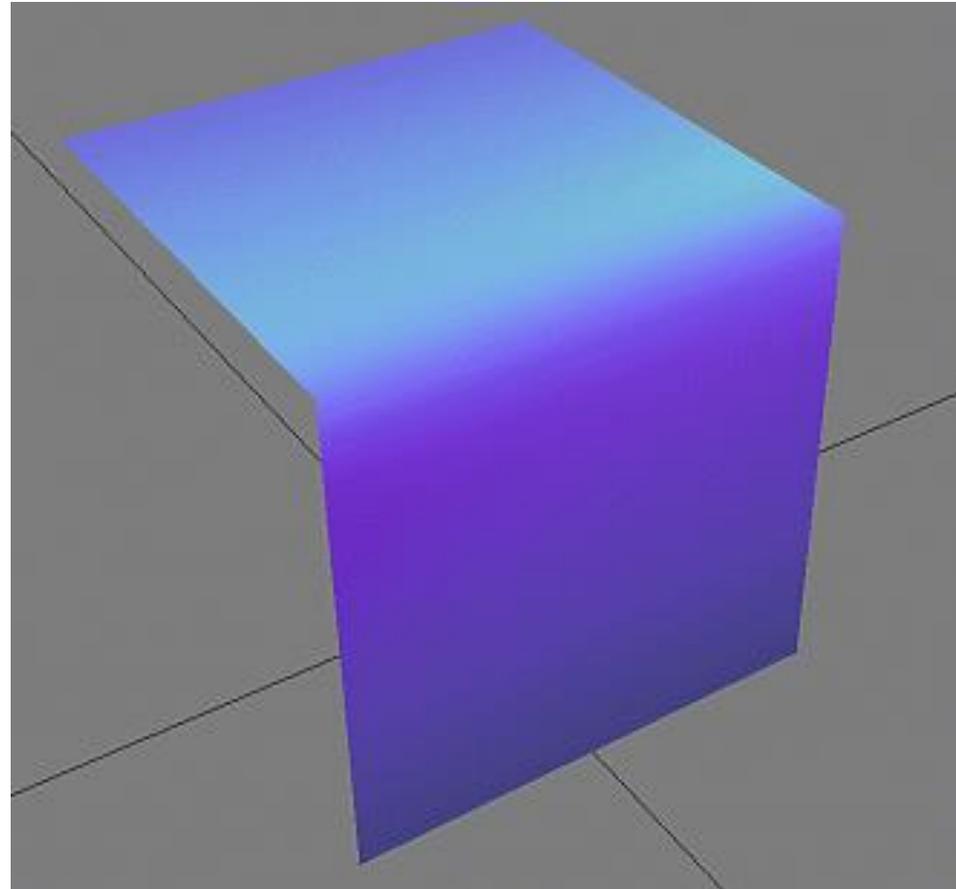


Нормал-маппинг. Нормали высокополигонального объекта

Получаем нормали
высокополигонального
объекта

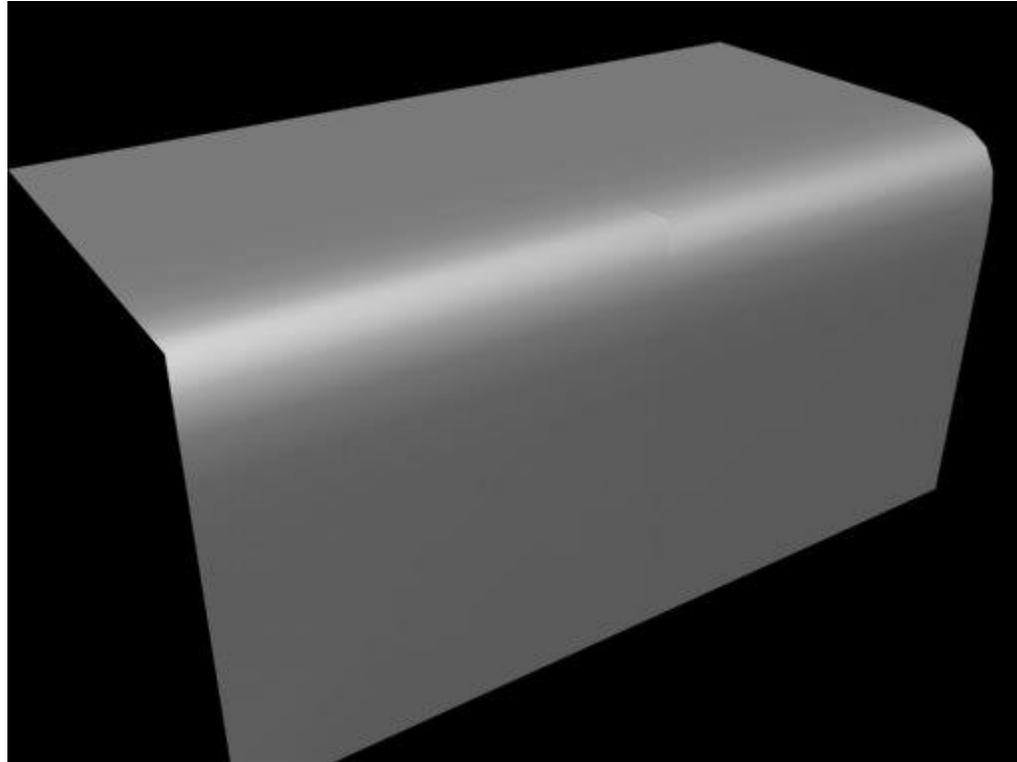


Нормал-маппинг. Карта нормалей как текстура

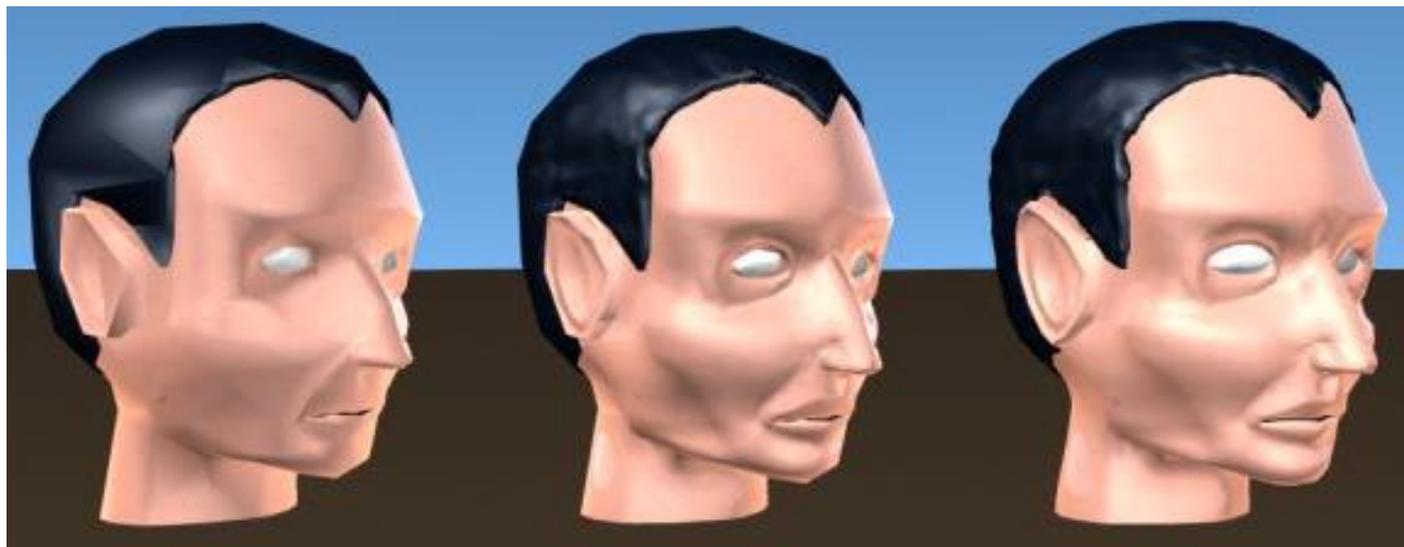


А вот как выглядит готовая карта нормалей.
Это обычное изображение, но в RGB-каналах закодированы векторы (X, Y, Z).

Нормал-маппинг. Результат

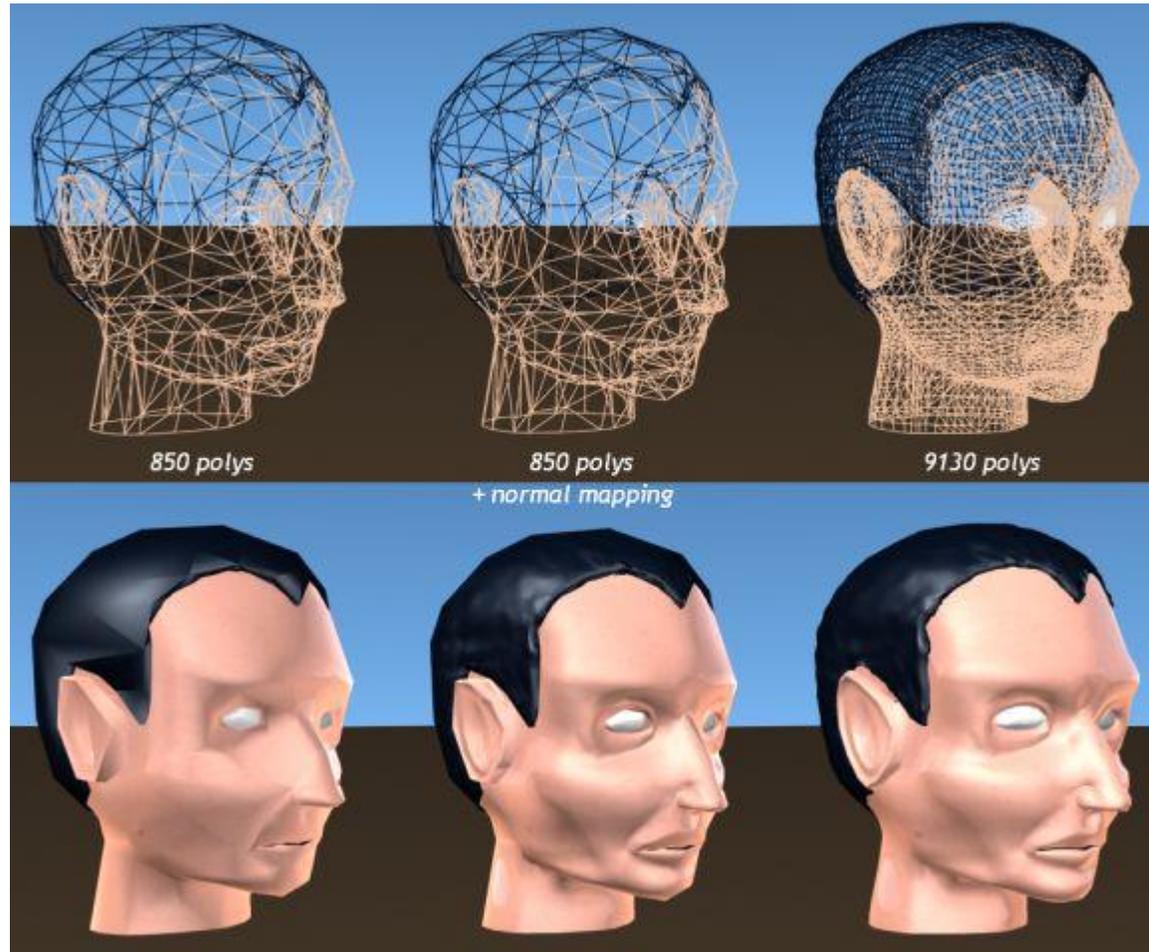


Где Normal Mapping?



Попробуйте определить

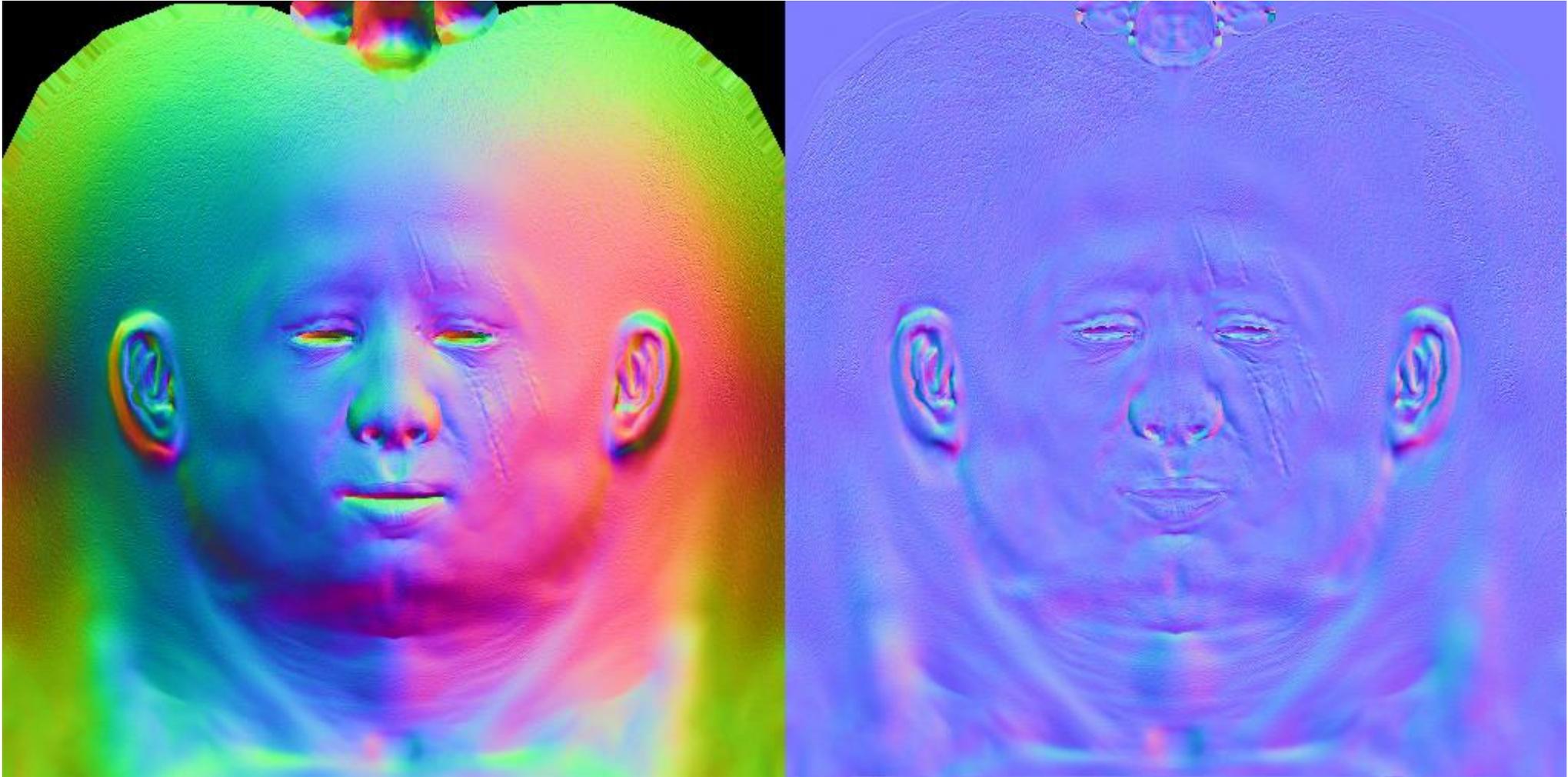
Normal Mapping: количество полигонов



Два вида карт нормалей

- В объектном пространстве (object-space): общей системе координат.
Здесь нормали заданы в единой системе координат для всего объекта.
Это хорошо для статичных объектов, которые не деформируются: стены, пол, мебель.
- В касательном пространстве (tangent-space): локальной системе координат.
Здесь нормали заданы относительно поверхности каждого треугольника.
Это позволяет объекту деформироваться — персонаж может двигаться, и нормали будут «приклеены» к поверхности.
Для анимированных персонажей используется только tangent-space.

Определите, где какая карта нормалей?



Общая и локальная система координат. Где какая?



Входные данные для вершинного шейдера

- Позиция камеры
- Позиция источника света

Алгоритм для вершинного шейдера

- Вычислить вектор источника света
- Нормализировать его
- **Трансформировать вектор источника света в пространство касательных.** Для этого нам нужна матрица, построенная из векторов T (касательная), B (бикасательная) и N (нормаль)
- Вычислить вектор камеры
- Нормализировать его
- Просчитать H вектор (модель освещения Блинна; H — half-angle vector)
- **Трансформировать H вектор в пространство касательных**

Выходные данные для вершинного шейдера

Вектор источника света в пространстве касательных

N вектор в пространстве касательных

Входные данные для фрагментного шейдера

- Вектор источника света в пространстве касательных
- N вектор в пространстве касательных
- Color — цвет

Алгоритм для фрагментного шейдера

- Нормализировать вектор источника света и N вектор в пространстве касательных
- Прочитать нормаль из `normal` текстуры и нормировать
- Посчитать скалярное произведение вектора нормали из `normal` текстуры и вектора источника света в пространстве касательных
- Просчитать скалярное произведение вектора нормали из `normal` текстуры и N вектора в пространстве касательных
- Просчитать цвет

Главный принцип применения нормалмаппинга

Низкополигональная модель должна сохранять основную форму объекта.

Нельзя взять куб и сделать из него сферу с помощью карты нормалей — крупные формы должны быть в геометрии.

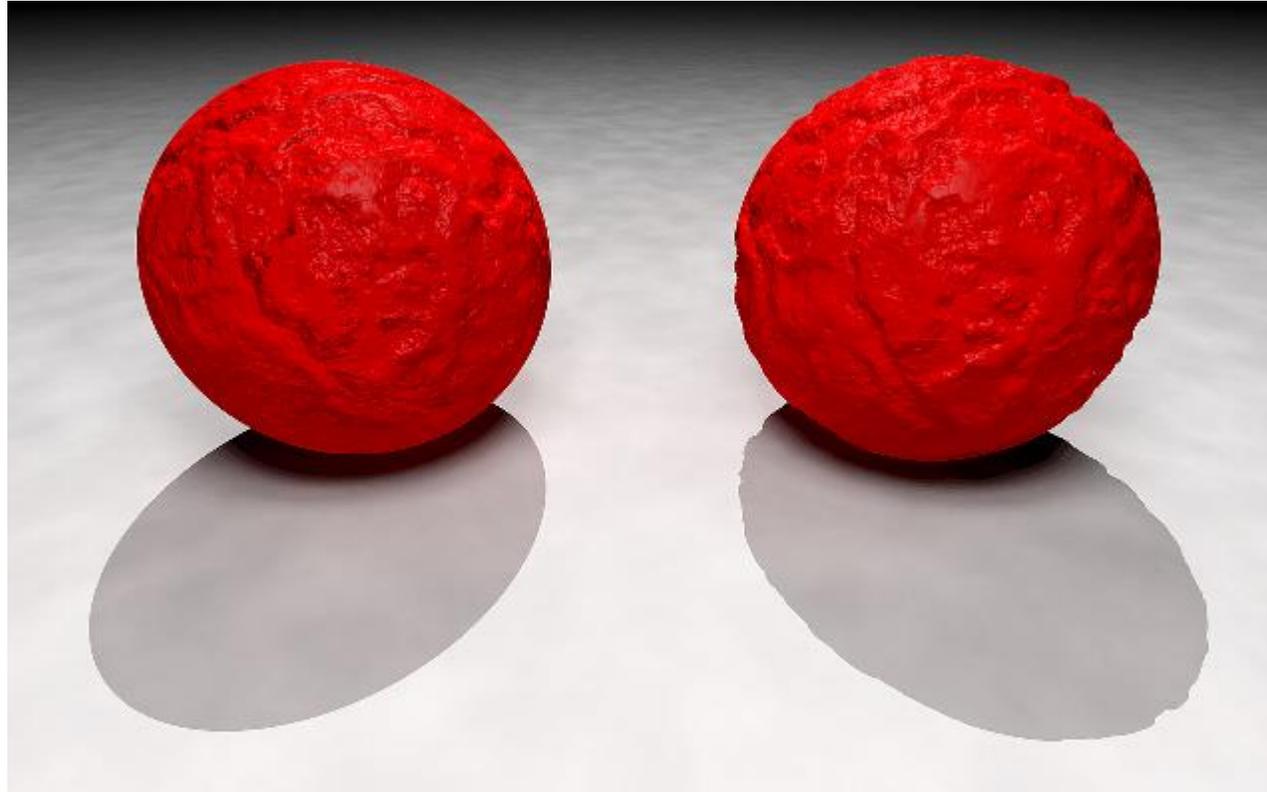
А вот царапины, морщины, поры, текстуру ткани — это можно и нужно добавлять через карты нормалей

Применение бампмаппинга

Для очень мелких деталей, которые даже в высокополигональной модели не моделируют (например, поры кожи), bump map может быть эффективнее и проще

Но в современных движках обычно используют normal map как более универсальное решение

Bump mapping и Displacement mapping



На левой картинке видна иллюзия объема, но силуэт остается плоским.

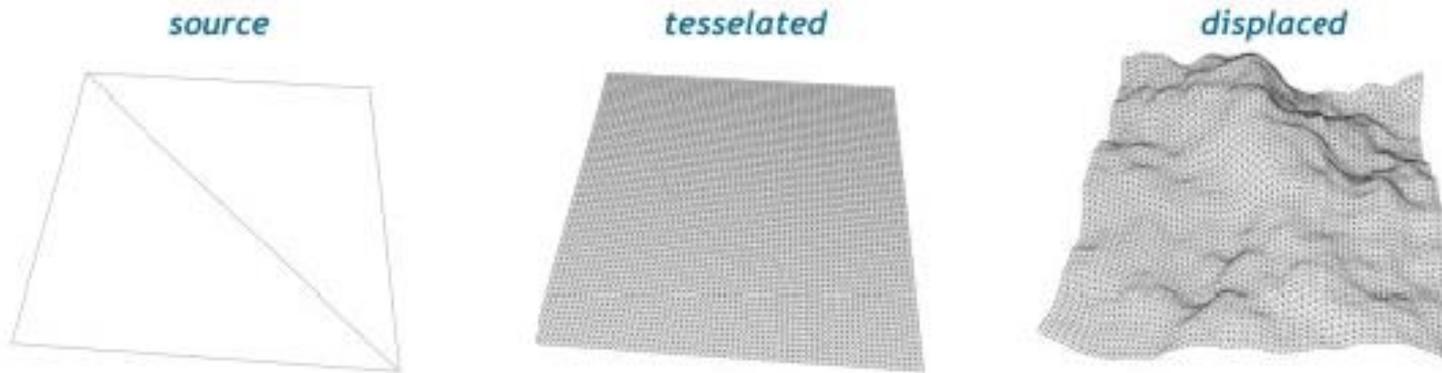
На правой — геометрия реально изменена, и край объекта становится неровным, как и должен быть у настоящего камня. В этом ключевое различие

Displacement mapping

- В отличие от bump и normal mapping, этот метод реально **изменяет геометрию**. **Вершины сдвигаются вдоль нормали на величину, считанную из карты высот.**
- Раньше это использовалось редко из-за производительности, но с современными технологиями тесселяции становится все популярнее.
- Карта смещения (displacement map) — это обычно полутоновая текстура, и значения в ней используются для определения высоты каждой точки поверхности объекта (значения могут храниться как 8-битные или 16-битные числа), схоже с bumpmap.

Создание ландшафта

- Часто карты смещения используются (в этом случае они называются и картами высот) для создания земной поверхности с холмами и впадинами
- Исходными были 4 вершины и 2 полигона, в итоге получился полноценный кусок ландшафта



Это основа практически всех игровых миров с процедурной генерацией

Сложный 3D объект



Позволяет достичь уровня детализации, невозможного при ручном моделировании.

Сжатие геометрии

- Интересный взгляд на displacement mapping — это метод сжатия геометрии.
- Вместо хранения миллионов треугольников мы храним простую сетку и карту высот.
- Это снижает требования к памяти и пропускной способности.
- Технология получила аппаратную поддержку начиная с DirectX 9.0.

Преимущества

- Позволяет использовать намного более сложные геометрически 3D модели
- Позволяет ускорить анимацию моделей
- Улучшить внешний вид модели, применив более сложные комплексные алгоритмы и техники, вроде имитации тканей (cloth simulation)

Ограничения

- Гладкие объекты, не содержащие большого количества тонких деталей, будут лучше представлены в виде стандартных полигональных сеток или иных поверхностей более высокого уровня, вроде кривых Безье
- Очень сложные модели, такие как деревья или растения, также нелегко представить картами смещения
- Почти всегда требует специализированных утилит, ведь очень сложно напрямую создавать карты смещения (если речь не идет о простых объектах, вроде ландшафта)

Parallax mapping

- Parallax mapping был представлен Томомити Канэко (Tomomichi Kaneko) в 2001 году www.researchgate.net/publication/228583097_Detailed_shape_representation_with_parallax_mapping
- В 2004 году Уэлш (Welsh) продемонстрировал применение параллаксмаппинга на программируемых видеочипах
- Parallax mapping полностью выполняется на графических процессорах видеокарты как пиксельный шейдер

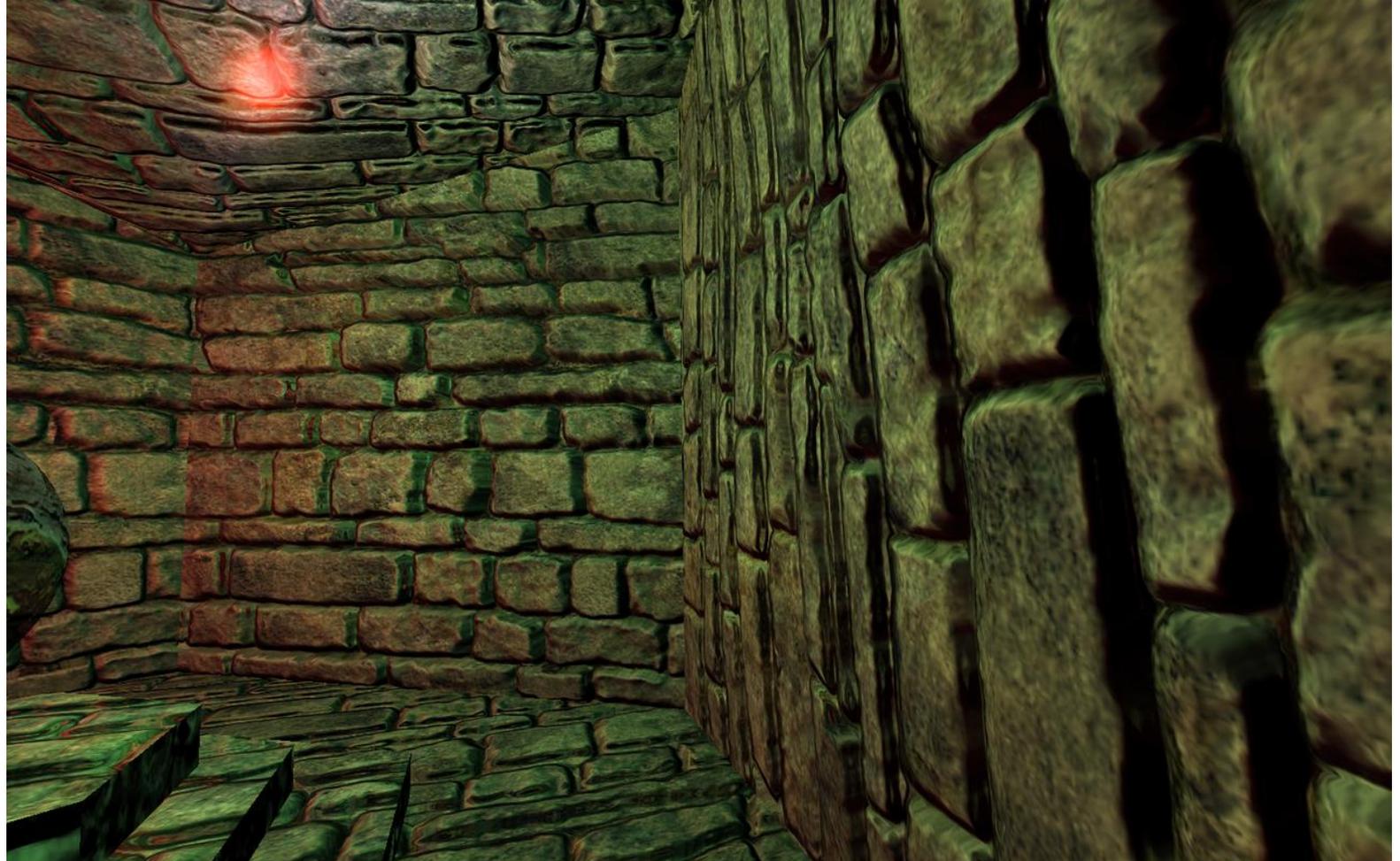
Другие названия

- Parallax Mapping
- Offset Mapping
- Virtual Displacement Mapping
- Per-Pixel Displacement Mapping

Все они описывают одну идею — смещение текстурных координат для имитации объема

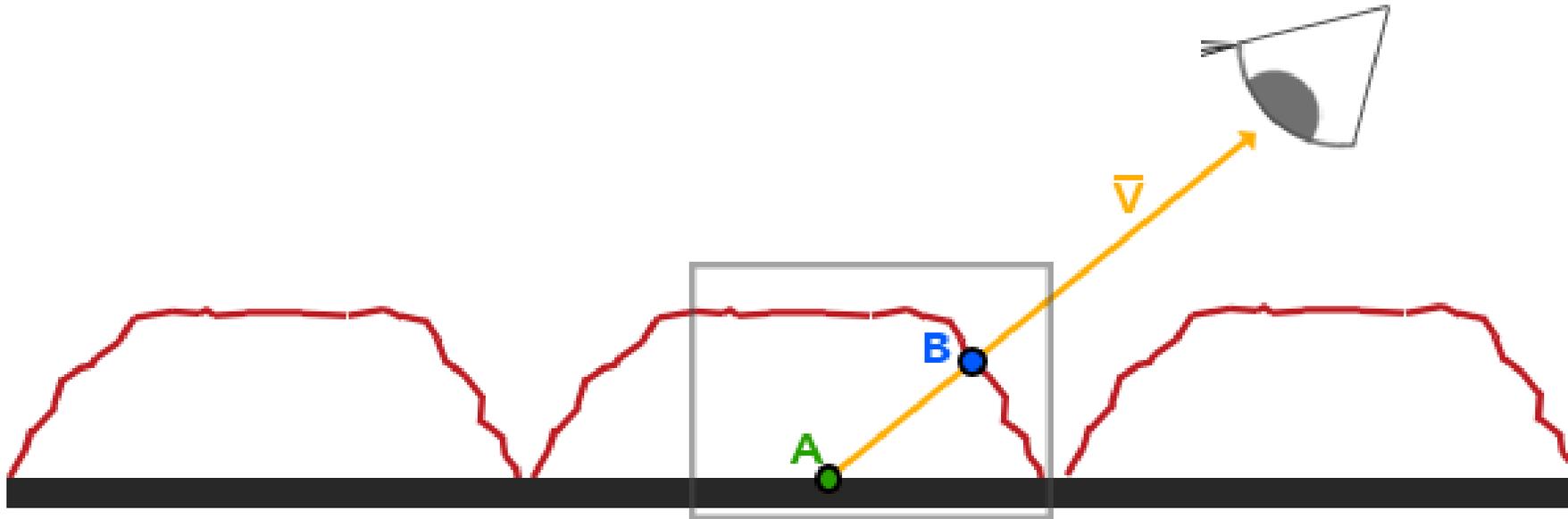
Parallax mapping

Parallax mapping
осуществляется
смещением текстурных
координат так, чтобы
поверхность казалась
объёмной



Скриншот из демонстрационного бенчмарка графического движка Irrlicht Engine

Идея Parallax mapping



Идея метода состоит в том, чтобы возвращать текстурные координаты той точки, где видовой вектор пересекает поверхность

Это требует просчета лучей (рейтрейсинг) для карты высот

Основная сложность

Как вычислить текстурные координаты точки В находясь в точке А?

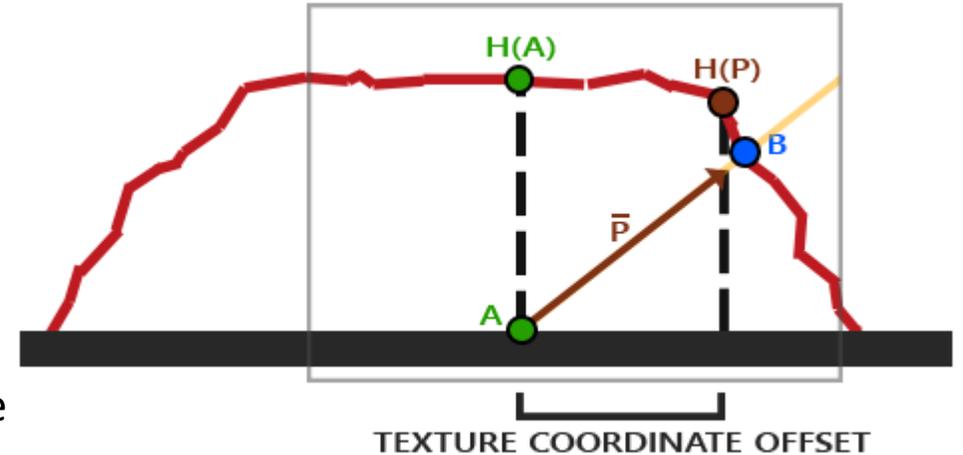
Parallax Mapping предлагает приближенное решение, применяя простое масштабирование вектора направления V от поверхности к наблюдателю на величину высоты для фрагмента А.

Т.е. просто меняем длину V так, чтобы она соответствовала величине выборки из карты высот $H(A)$, соответствующей фрагменту А.

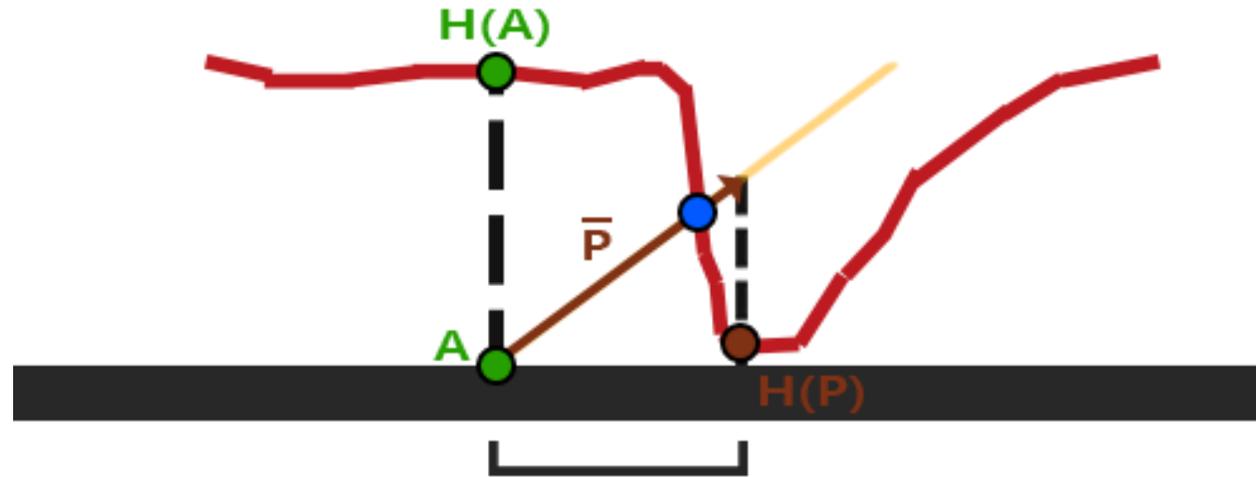
Вектор P – результат масштабирования

Далее результирующий вектор P раскладывается на компоненты в соответствии с системой координат самой плоскости, которые используются как смещения для исходных текстурных координат.

При этом, поскольку вектор P вычисляется с использованием величины из карты высот, то чем больше значение высоты соответствует текущему фрагменту, тем сильнее для него будет смещение



Проблемы



Этот простой прием дает неплохие результаты в ряде случаев, но все же является очень грубой оценкой положения точки В.

Если карта высот содержит участки с резко меняющимися значениями, то результат смещения становится некорректным: скорее всего вектор P даже близко не будет попадать в окрестность точки В

Parallax mapping и аппроксимация

- Если карта высот не имеет слишком сильно изменяющихся значений («гладкая» или «плавная»), то можно обойтись аппроксимацией без использования рейтрейсинга
- Parallax mapping условно можно назвать «2.5D»

Displacement mapping и Parallax mapping

Главное отличие **parallax mapping** от **displacement mapping** в том, что в нём все расчеты **попиксельные**, а не **повершинные**

Normal mapping и Parallax mapping

- Данная технология также использует карты нормалей
- В отличие от normal mapping, она реализует не только освещение с учётом рельефа, но и сдвигает координаты диффузной текстуры
- Этим достигается наиболее полный эффект рельефа, особенно при взгляде на поверхность под углом
- Parallax mapping отличается от normal mapping всего тремя инструкциями пиксельного шейдера: две математические инструкции и одна дополнительная выборка из текстуры
- После того, как вычислена новая текстурная координата, она используется дальше для чтения других текстурных слоев: базовой текстуры, карты нормалей и т. п

Применение в играх — пример 1

Parallax mapping активно использовался в играх середины 2000-х



Применение в играх — пример 2



Недостатки Parallax mapping

- Использование обычного параллакса ограничено картами высот с небольшой разницей значений.
- «Крутые» неровности обрабатываются алгоритмом некорректно, появляются различные артефакты, «плавание» текстур и пр.

Модификации Parallax mapping

Поэтому исследователи предложили улучшенные версии:

Иерекс, Доннелли, Татарчук, Поликарпо (Yerex, Donnelly, Tatarchuk, Policarpo) — каждый внес свой вклад.

Общая идея — использовать трассировку лучей в пиксельном шейдере, чтобы более точно определять пересечения.

Эти методы получили названия Parallax Mapping with Occlusion, Parallax Mapping with Distance Functions, и самый известный — Parallax Occlusion Mapping.

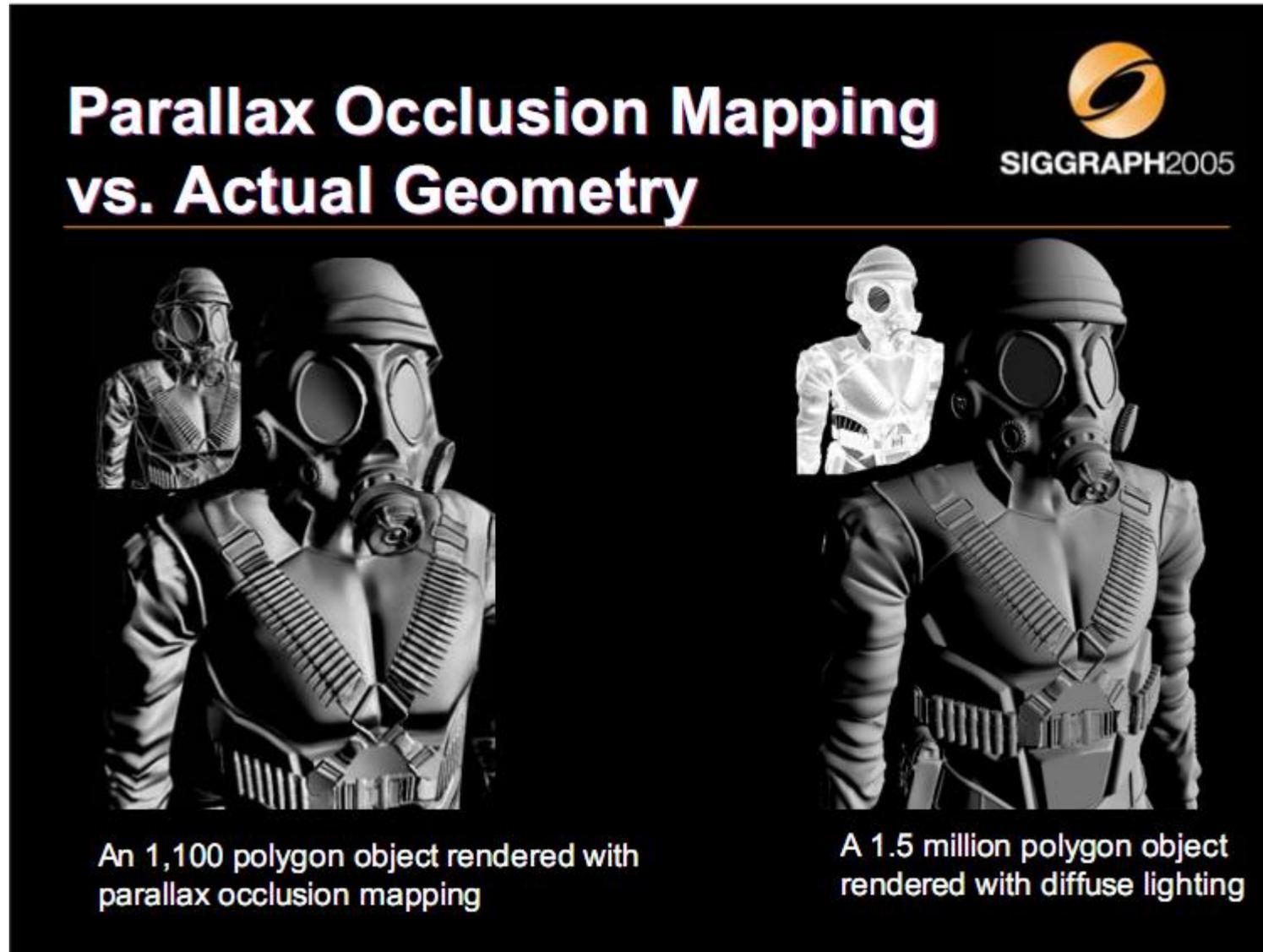
Parallax occlusion mapping

- Parallax occlusion mapping, или POM — это усовершенствованная техника, которая позволяет корректно обрабатывать самозатенение и перспективу.
- Вместо одного шага аппроксимации она делает несколько шагов по лучу, ища реальное пересечение с поверхностью, заданной картой высот.

История POM (Parallax occlusion mapping)

- Первая работа по POM появилась в 2004 году в книге ShaderX3. Авторы — Зоя Броули и Наталия Татарчук. Затем, в 2005, на SIGGRAPH, Татарчук представила развитие техники.
- Компания ATI использовала POM для демонстрации возможностей Shader Model 3.0 в новейшей на то время видеокарте Radeon X1800.
- Первым игровым движком, в котором использовался Parallax occlusion mapping, стал CryEngine 2 от немецкого разработчика Crytek, который впервые использовался в компьютерной ПК-игре Crysis 2007 года выпуска

Из доклада Наталии Татарчук



Из доклада Наталии Татарчук

Parallax Occlusion Mapping vs. Actual Geometry



An 1,100 polygon object rendered with parallax occlusion mapping (wireframe)

A 1.5 million polygon object rendered with diffuse lighting (wireframe)

Из доклада Наталии Татарчук



SIGGRAPH2005

Parallax Occlusion Mapping vs. Actual Geometry

	<ul style="list-style-type: none">- 1100 polygons with parallax occlusion mapping (8 to 50 samples used)- Memory: 79K vertex buffer 6K index buffer 13Mb texture (3Dc) (2048 x 2048 maps) <hr/> <p>Total: < 14 Mb</p>	<p>Frame Rate:</p> <ul style="list-style-type: none">- 255 fps on ATI Radeon hardware- 235 fps with skinning
	<ul style="list-style-type: none">- 1,500,000 polygons with diffuse lighting- Memory: 31Mb vertex buffer 14Mb index buffer <hr/> <p>Total: 45 Mb</p>	<p>Frame Rate:</p> <ul style="list-style-type: none">- 32 fps on ATI Radeon hardware

POM (Parallax occlusion mapping) в 3DMark Vantage



Реализация алгоритма Parallax occlusion mapping в бенчмарке 3DMark Vantage 2008 года. Данное изображение построено при помощи двух полигонов (треугольников), расположенных на одной плоскости

Parallax occlusion mapping

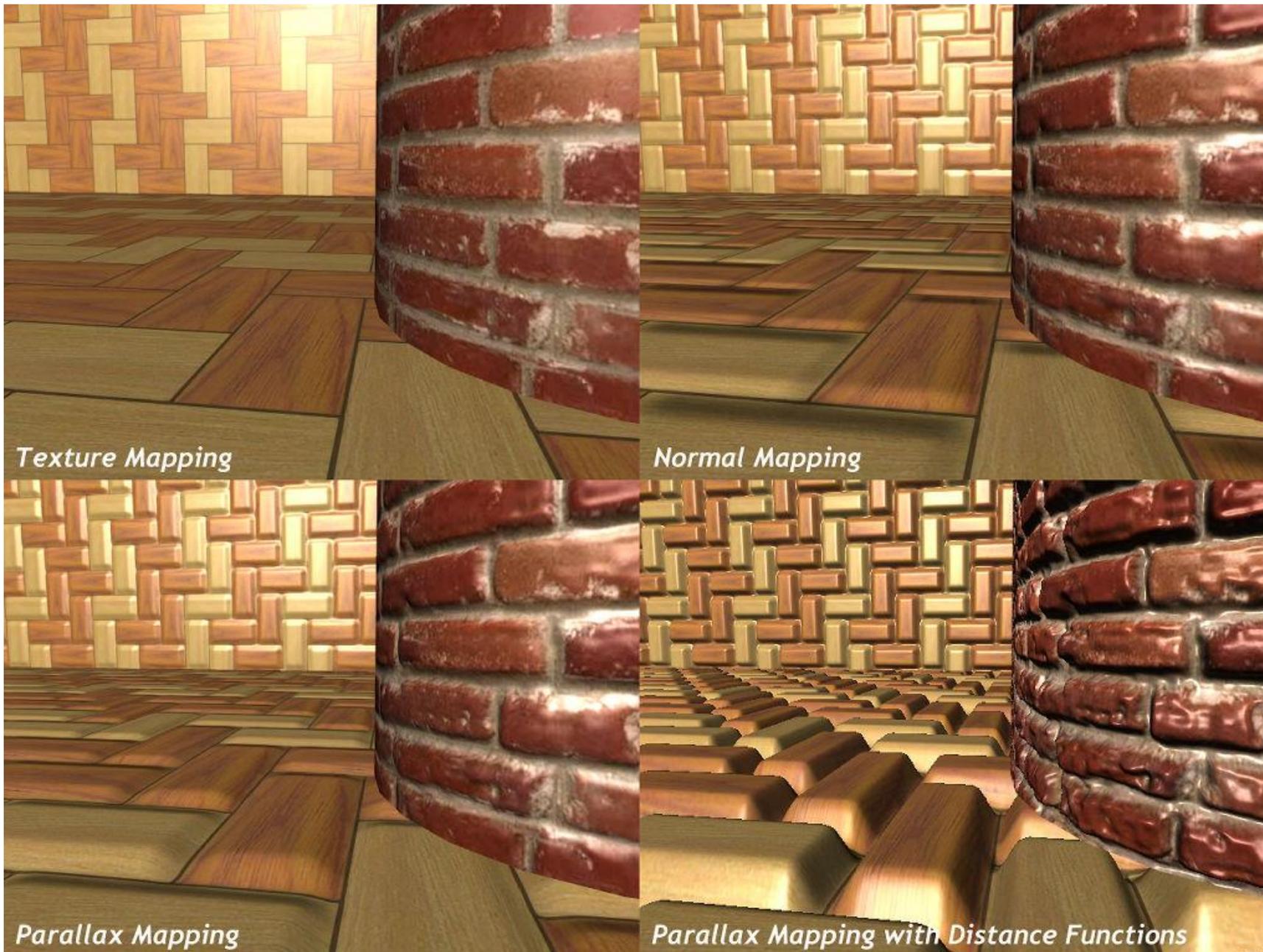


Недостатки

Невысокая детализация силуэтов и граней

Видеокарта должна обеспечивать надлежащий уровень скорости исполнения операций ветвления в пиксельном шейдере

Отсутствие геометрически правильных силуэтов (краев объекта)



Texture Mapping

Normal Mapping

Parallax Mapping

Parallax Mapping with Distance Functions

Метод	Суть	Когда применять
Bump mapping (1978, Блинн)	Возмущение нормалей по карте высот. Хранит высоту, вычисляет градиенты.	Простые поверхности, ограниченная детализация, legacy-проекты.
Normal mapping (1998)	Хранит готовые нормали в текстуре (RGB = XYZ). Два типа: object-space (статичные объекты) и tangent-space (анимированные).	Стандарт индустрии. Персонажи, окружение, объекты.
Displacement mapping (2002)	Реальное изменение геометрии: вершины сдвигаются по карте высот.	Ландшафты, крупные детали, требующие изменения силуэта.
Parallax mapping (2001)	Смещение текстурных координат по карте высот для имитации объема.	Плавные поверхности, где нужна дешевая иллюзия глубины.
Parallax Occlusion Mapping (POM) (2004)	Трассировка луча в пиксельном шейдере, учет самозатенения.	Максимальная детализация на плоской геометрии (Crysis, 2007).

Главный принцип

Крупные формы — в геометрии, мелкие детали — в текстурах.

Иллюзия объема создается освещением, а не полигонами.

Связь с курсом

Лекция 3 (Освещение) — bump/normal mapping модифицируют нормали, которые участвуют в модели Фонга.

Лекция 4 (Текстуры) — все методы используют текстуры как источник данных (карты высот, карты нормалей).

Лекция 5 (Рельефное текстурирование) — применение этих знаний для создания фотореалистичных поверхностей.

Вопросы для самопроверки

Базовый уровень

1. Чем bump mapping отличается от normal mapping?
2. В чем разница между object-space и tangent-space картами нормалей? Для чего нужен каждый тип?
3. Почему для анимированных персонажей нельзя использовать object-space normal maps?

Средний уровень

4. Какая главная проблема простого parallax mapping? При каких условиях он работает хорошо?
5. Что такое Parallax Occlusion Mapping (POM) и чем он лучше обычного parallax mapping?
6. В чем ключевое различие между displacement mapping и всеми остальными методами рельефного текстурирования?

Продвинутый уровень

7. Зная формулу $New_Normal = Normal + (U * x_gradient) + (V * y_gradient)$, объясните, зачем нужны векторы U и V и что произойдет, если их не учитывать?
8. Почему в POM силуэты объектов остаются гладкими? Как можно это исправить?
9. (Исследовательское) Найдите в любой современной игре (например, Cyberpunk 2077, The Last of Us Part II) сцену с каменной стеной или дорогой. Как вы думаете, какие методы рельефного текстурирования там использованы?